



Procedia Computer Science

Volume 29, 2014, Pages 1569–1579

ICCS 2014. 14th International Conference on Computational Science



Argumentation Approach and Learning Methods in Intelligent Decision Support Systems in the Presence of Inconsistent Data

Marina Fomina¹, Oleg Morosin¹, Vadim Vagin¹¹National Research University "Moscow Power Engineering Institute"
Moscow, Russian Federation

m_fomina2000@mail.ru, omorsik@gmail.com, vagin@appmat.ru

Abstract

This paper contains a description of methods and algorithms for working with inconsistent data in intelligent decision support systems. An argumentation approach and application of rough sets for generalization problems are considered. The methods for finding the conflicts and the generalization algorithm based on rough sets are proposed. Noise models in the generalization algorithm are viewed. Experimental results are introduced. A decision of some problems that are not solvable in classical logics is given.

Keywords: argumentation, inconsistent data, rough sets, noisy data, generalization algorithms, notion formation

1 Introduction

The problem of construction of intelligent decision support systems (IDSSs), orientated towards open and dynamic applied domains is currently very important in AI. Unfortunately, the most of practical problems are rather ill formalized, and may contain inconsistent, uncertain and implausible information. The methods of the classical logics couldn't be applied for solving such problems. To cope with such problems it is necessary to use the methods of plausible reasoning that allow to find some acceptable (non-necessarily optimal) solution. The key point of this paper is to present some methods and approaches to deal with inconsistent and noisy (raw) databases (DB) used for the inductive notion formation. Argumentation approach is proposed for processing inconsistent data in knowledge bases (KBs). Generalization algorithms, based on the rough set theory for the inductive notion formation in the case of noisy data, are presented.

The paper is structured as follows. In section 2, we consider the unit structure of a typical IDSS and propose some methods to treat inconsistent information. In section 3, we formulate argumentation approach for solving problems in the presence of inconstant data. In section 3.1, defeasible reasoning as argumentation formalization is discussed. In section 3.2 methods for finding conflicts are presented and in section 3.3, methods for justification degrees of arguments are given. Further in section 4

including 4.1, learning methods in the presence of inconsistent data, the generalization problem and methods based on the rough set theory are viewed. In section 4.2, basic concepts of the rough set theory are given. In section 4.3, two noise models are considered. And in section 4.4, the results of computer modeling performed on UCI ML Repository with an entering of noise in learning sets are presented. Conclusions are given in section 5.

2 General Architecture of IDSS

One of actual problems in the AI area is a problem of the construction of intelligent systems, typical representative of which is an IDSS oriented towards open and dynamic applied domains [1].

IDSSs are based on the integration of knowledge representation and manipulation models that are capable to adaptation, modification, and learning. Such models are oriented to specific problem domains that reflect the ability to set up and modify their states.

The generalized structure of an IDSS is shown in Fig. 1.

When implementing methods of reasoning in IDSSs one should take into consideration the following features of such systems:

- the necessity to take a decision under constraints defined by an actually controlled process;
- the impossibility of obtaining all the objective information necessary for decision making and in this connection the usage of subjective expert information;
- the multivariate character of search;
- the necessity of applying methods of plausible reasoning and the active participation of a decision making person (DMP) in decision making processes ;
- the presence of incomplete, fuzzy and even inconsistent data for situation description.

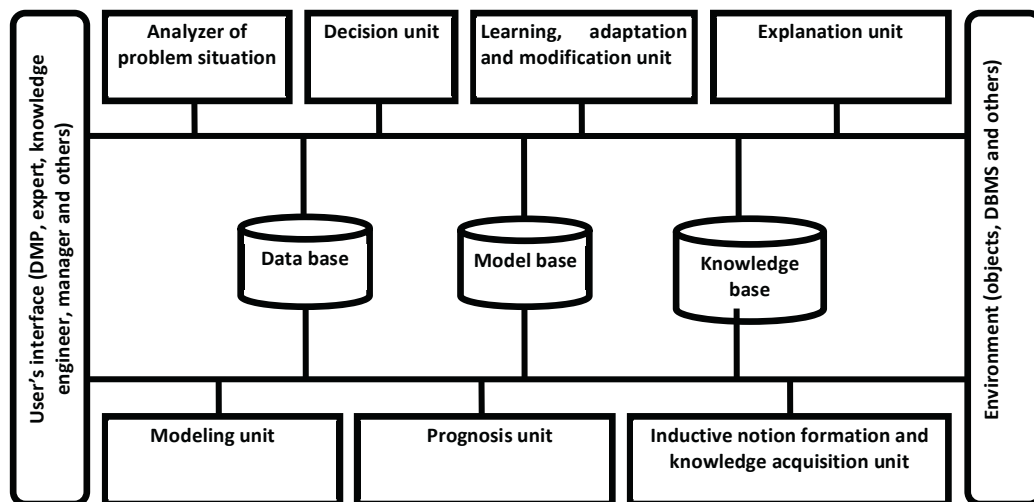


Figure 1: General architecture of an IDSS

Models of knowledge representation in IDSSs have some special peculiarities.

First of all let's consider what types of information could be used in IDSSs and define what could lead to inconsistency. As usual, the following types of information are used in intelligent systems: objective, subjective and hypothetical information [2].

1. Objective information - information obtained from reliable sources, or which can be directly measured or validated. Such information is generally used as facts and considered as reliable.

2. Subjective information - information derived from less reliable sources. These may be some assumptions or judgments. Often they are formulated using the phrases "typically," "usually," "likely". Such information serves as a "source" of contradictions and conflicts. In case this information is received from some statistical data, it can have some measure of plausibility. However, as usual, there is no source of obtaining any certain degree of justification and therefore the problem of handling uncertain and qualitative information is very important.

3. Hypothetical information. This type of information is necessary for the construction of hypotheses. It is not necessary for hypothetical information to be true. The absence of needed information is a reason of building hypotheses that should be proven later. Thus, construction of hypotheses is a necessary part of any IDSS.

Any IDSS that uses subjective or hypothetical information (for instance, the expert knowledge) would have problems with treating such information in the decision unit, as well as in the inductive notion formation and the knowledge acquisition unit.

A decision unit is intended for implementing methods of solving problem (using argumentation approach) with inconsistent data while notion generalization methods with noisy and inconsistent data are realized in the inductive notion formation and knowledge acquisition unit.

3 Applying the Argumentation Theory for Solving Problems in Conditions of Inconsistent Data

In decision support systems which use the first order predicate logic as a knowledge representation model, there are usually a number of restrictions, that guaranties the absence of inconsistencies in KBs. Classical methods of inference don't have mechanisms of 'revision' of previously made conclusions. Also there no mechanisms for detection and solving conflicts. We propose to use the mechanism of argumentation for dealing with conflicts and inconsistency. Argumentation gives us much more instruments for modelling plausible reasoning. There are several formalizations of the argumentation theory. For instance, there are the abstract argumentation system, proposed by P.M Dung [3], the argumentation system of F. Lin and Y. Shoham [4], the G.A.W. Vreeswijk's system [5] and systems based on defeasible reasoning [6] and some others.

We'll consider the development of defeasible argumentation systems that use the first order logic as an underlying language [7], and based on the theory of defeasible reasoning proposed by J. Pollock [6].

At first, it is necessary to introduce the basic definitions and notions. **Argumentation** is usually considered as the process of considering arguments "for" and "against" some assumption. In contrast to abstract argumentation, defeasible reasoning provides mechanism for obtaining new arguments from existing ones. We will consider the **argument** as a pair consisting of a set of premises and a conclusion [6]. Such couples will be written so: p / X , where p is a conclusion, and X - a set of premises. For example, the argument $(p \rightarrow q) / \{\sim A, B\}$ means that premises $\sim A, B$ leads to the conclusion $p \rightarrow q$. All interrelations between arguments will be represented on an **inference graph**. It is a graph that shows the way of building new arguments from already existing ones. As well, the inference graph shows the conflicts between arguments.

As it was already mentioned above, conflicts appear because of inconsistent data in a KB. Detection and solving of such conflicts is the main point of an argumentation system. We will consider two types of conflicts – *rebutting* and *undercut* [8].

Rebutting is a situation when some of the arguments rebut the conclusion of the other arguments. In other words, the argument $A_1 = p_1 / X_1$ rebuts the argument $A_2 = p_2 / X_2$, when the conclusion p_1 rebuts conclusion p_2 and $X_1 \in X_2$. Rebutting is a symmetrical form of attack.

Undercut is the asymmetric form of attack when one argument undercuts a link between premises and the conclusion of the other argument.

It is necessary to note, that in contrast to rebutting, there are no mechanisms for constructing such conflicts automatically, so they must be formulated manually, and entered by a user. So it is necessary to define rules of the formulization of such type of conflicts.

3.1 The Method for Finding Conflicts

Detection of conflicts is the most difficult part in defeasible reasoning. Main difficulties arise from necessity of maintenance of the first order logic. For the propositional logic, it is clear that rebutting is a situation, when two contradicting arguments A and $\sim A$ exist. Speaking of the first order logic, the main idea is the application of a unification mechanism for finding rebutting arguments. A substitution $U = \{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}$, where t_i – term and x_i – variable, is called the *unifier* for expressions W_1 and W_2 if $W_1 \cdot U = W_2 \cdot U$, where $W_1 \cdot U$ is the result of replacing variables x_1, x_2, \dots, x_n on terms t_1, t_2, \dots, t_n and it is considered that W_1 and W_2 do not have variables with the same names (if so it is necessary to rename them) [9]. For example, two expressions $P(x_1) \& G(x_2) \rightarrow H(x_1, x_2)$ and $P(f(a)) \& G(b) \rightarrow H(f(a), b)$ are unifiable with the unifier $U = \{f(a)/x_1, b/x_2\}$, and expressions $P(x) \rightarrow G(x)$ and $P(a) \& G(b)$ are not unifiable.

Two arguments $A_1 = p_1 / X_1$ and $A_2 = p_2 / X_2$ have a conflict of the type “rebutting” if exists such unifier U that:

$$\sim p_1 \cdot U = p_2 \cdot U, \text{ where } \sim p_1 \text{ is the negation of the conclusion } p_1.$$

$$X_1 \cdot U \in X_2 \cdot U \text{ or } X_2 \cdot U \in X_1 \cdot U.$$

An argument A_1 is the undercutting argument for arguments A_2 and A_3 if:

- there is a defeasible link between A_2 and A_3 ;
- there is the undercutting rule $E \square (C @ D)$ and there are unifiers U_1 and U_2 such that:

$$E \cdot U_1 = A_1 \cdot U_1$$

$$(C \cdot U_1) \cdot U_2 = (A_2) \cdot U_2$$

$$(D \cdot U_1) \cdot U_2 = (A_3) \cdot U_2.$$

The proposed methods for finding conflicts were successfully implemented in the algorithm of conflicts detection [10].

3.2 Justification Degrees in Defeasible Reasoning

First of all it is necessary to define how this degrees could be obtained and the way they are entered. Here we will use a numerical scale $[0, 1]$ for assigning justification degrees [11]. We assume that 0 means that there are no points for believing the argument is valid (it means that the argument is defeated) and 1 means that the argument is totally undefeated, that there no reasons to doubt it is true. There could be two types of justification degrees:

- justification degrees of initial arguments;
- justification degrees of defeasible rules of inference.

The first type of degrees is assigned to an every initial argument and could be considered as the degree of reliability of the source it is obtained from. For instance, the forecast is that the chance of raining is 70%. So an argument $A1: \text{Tomorrow}(\text{raining})$ with justification degree 0.7 could be made. Let's define a function $Jus(A)$ for a justification degree of an argument. The exact mechanisms of obtaining the justification degrees mostly depend on an applied domain. For instance, this could be statistical data or some expert judgments (e.g. the possibility that company's shares will grow is 60%).

The other type of justification degrees is related with defeasible rules of inference. As it was already mentioned above, defeasible rules are often a formalization of expert knowledge of the

following type: “If A is true, than as usual B is also true”. Such rules of inference could also have some numerical degree of justification. The application of both types of degrees is rather a difficult task and needs to some further investigations. So in this paper we’ll limit the observation of only the first type of degrees.

It is necessary to define function $Jus(A)$ that will calculate the justification degree of an every argument in an inference graph. We will consider that it is known for an every initial argument. The value of this function for an exact argument is influenced by two factors: the tree of inference (i.e. the justification degrees of arguments, which were used in the inference of that argument) and conflicts with other arguments. For the sake of convenience, let’s consider these two factors separately: $Jus_{anc}(A)$ – an inherited justification degree of ancestors and $Jus_{con}(A)$ – the value of influence of the conflict the argument has.

$$Jus_{anc}(A) = \min(\{Jus(A_1), Jus(A_2) \dots Jus(A_n)\}), \quad (3.1)$$

where A_1, A_2, \dots, A_n – arguments that were used in the inference of argument A .

Formula (3.1) is called the weakest link principle [6]. It is necessary to note, that this is not the only way of calculating justification degree, e.g. in some works, Bayesian approach is applied [12].

From formula (3.1) it follows that if to perform calculation of justification degrees recursively, it is possible to search for the minimum among the direct ancestors that were used in a previous step. In the particular case, when the argument has only one ancestor, its Jus_{anc} will be the same as its ancestor degree.

On the contrary to calculating Jus_{anc} (when the weakest arguments are considered), calculation of $Jus_{con}(A)$ is based on finding the conflict with the argument, that have the highest degree of justification. Let’s consider $Aconfl$ as a set of arguments that has conflicts with the argument A and $Aconfl_i$ is a member of this set.

$$Jus_{con}(A) = \begin{cases} \max(\{Jus_{anc}(Aconfl_1), Jus_{anc}(Aconfl_2), \dots, Jus_{anc}(Aconfl_n)\}), & |Aconfl| > 0; \\ 0, & otherwise. \end{cases} \quad (3.2)$$

In formula (3.2) Jus_{anc} is used to avoid cases of mutual conflicts between two arguments.

Thus, the justification degree of an argument A could be calculated as follows:

$$Jus(A) = \begin{cases} Jus_{anc}(A) - Jus_{con}(A), & Jus_{anc}(A) > Jus_{con}(A); \\ 0, & otherwise. \end{cases} \quad (3.3)$$

It was shown how inconsistent data could be treated in the decision unit. Further we will consider some methods that could be implemented in the inductive notion formation and knowledge acquisition unit.

4 Learning Methods in the Presence of Noisy and Inconsistent Data

Another important component in every IDSS is the inductive notion formation and the knowledge acquisition unit. Using generalization methods in intelligent systems, it is possible to find main patterns that are mostly typical for different situation classes, which appear in a complex system. It is made by performing analyze of already classified objects and building some generalized rules (so called, a generalized model). Then this generalized model can be already used to recognize situations unknown before.

The purpose of this section is to examine the possibility of using rough sets theory for inductive notion formation in conditions of inconsistent and incomplete data.

4.1 Generalization problem

By search the solution in IDSSs, it is reasonable to apply the plausible inference methods that allow to find an applicable solution (that is maybe not optimal). One of the approaches is based on the fact that at the first stage of the solution search of a new unknown problem, a person (expert or DMP) tries to use decisions that were taken previously in similar cases and, if necessary, adapt them to the problem (the current situation). The approach based on forming inductive descriptions for classes of similar situations is very important. Such approach is related with solving the problem of inductive notion formation or the generalization problem.

Let us give the formulation of feature-based concept generalization [9]. Let $O = \{o_1, o_2, \dots, o_n\}$ be a set of objects. Each object can represent one situation which is characterized by k attributes: a_1, a_2, \dots, a_k . Denote by $Dom(a_1), Dom(a_2), \dots, Dom(a_k)$ the sets of admissible values of features where $Dom(a_i) = \{c_1, c_2, \dots, c_{q_k}\}$, $1 \leq i \leq k$, and q_k is the number of different values of the feature a_i . Such a description of an object is called a feature description. Quantitative, qualitative, or scaled features can be used as object features [9]. Let O be the set of all situations represented in a certain IDSS. There is the set V of situations, in which identical or similar decisions were accepted. We can call V as a class of situations. All situations included in V form a set of positive objects related to some class (concept) and let W be a set of negative objects. We will consider the case where $S = V \cup W$, $V \cap W = \emptyset$. Let U be a non-empty set of objects such that $U \subset V \cup W$. We call U a learning set. It is necessary to build a rule, based on a learning set that separates positive and negative objects of a learning set.

Thus, the concept is formed if one manages to build a decision rule which, for any example from a learning sample, indicates whether this example belongs to the concept or not. The proposed algorithm generates a decision in the form of rules of the type "IF <condition>, THEN <the desired concept>". The condition is represented in the form of a logical function in which the Boolean variables reflecting the feature values are connected by logical connectives. Further, instead of the notion "feature" we will use the notion "attribute". The decision rule is correct if later it successfully recognizes objects which originally did not belong to the learning sample. The information about real objects may be imperfect, uncertain and incomplete [13]. Real data sets may contain noise. The presence of noise in data changes the above mentioned setting up of the generalization problem both at the stage of building decision rules and at the stage of the object classification. First of all, the original learning sample U is replaced by the sample U' in which distorted values or missing values of attributes occur with a certain probability.

We consider the solution of the concept generalization problem using the methods of rough set approach.

4.2 Method of Generalization Based on Rough Sets

The rough set theory has been proposed in the beginning of 80th years of the last century by the Polish mathematician Z. Pawlak [14], [15]. Later this theory was developed by many researchers and applied to the decision of various tasks.

A rough set is defined by the assignment of upper and lower boundaries of a certain set called the approximations of this set. Each element of the lower approximation is certainly an element of the set. Each element that does not belong to the upper approximation is certainly not an element of the set. The difference in the upper and lower approximations of a rough set forms the so-called boundary region. The element of the boundary region is probably (but not certainly) an element of the set. Similarly to fuzzy sets, rough sets are mathematical notion for work with fuzziness in data.

We will consider how the rough set theory can be used to solve notion generalization problem.

In Pawlak's works [14], [15] the notion of an information system has been introduced. An *information system* is understood as pair $S = (U, A)$, where $U = \{x_1, x_2, \dots, x_n\}$ is a non-empty finite set of objects named *learning set* or *universe*, and $A = \{a_1, a_2, \dots, a_k\}$ is a non-empty finite *set of attributes*. A *decision table* (or *decision system*) is an information system of the form $S = (U, A \cup \{d\})$, where $d \notin A$ is a distinguished attribute called *decision* or *decision attribute*, and all $a_i \in A$ are conditional attributes.

Let us introduce the *indiscernibility* or *equivalence relation* on the learning set $U: IND(A) \sqsubseteq U \sqsubseteq U$. We will say, that if $(x, y) \in IND(A)$ then x and y are indiscernible by values of attributes from A . A set of equivalence classes of relation $IND(A)$ is denoted by $\{X_1^A, X_2^A, \dots, X_m^A\}$. Then we can approximately define set X using attribute values by constructing of lower and upper approximations of X , designated by $\underline{A}X$ and $\overline{A}X$ respectively.

Since not all conditional attributes are equally important, some of them can be excluded from a decision table without loss of the information contained in the table. The minimal subset of attributes $B \subseteq A$ which allows to keep the generalized decision for all objects of learning set, is called *decision-relative reduct* of a table $S = (U, A \cup \{d\})$. Further, instead of a decision-relative reduct we will use the term *a reduct* [16].

Process of search of a reduct is a very important stage of any method using the rough set approach. For the search of reducts we propose the method, based on the following ideas:

- We consider as the significant attributes those attributes that are contained in intersection of all reducts of an information system.
- Building of dynamic reducts [16], i.e. building the conditional attribute sets appearing "sufficiently often" as reducts of subsamples of the original decision system. The attributes belonging to the "most" of dynamic reducts - approximated reduct - are considered as relevant. The thresholds of values for "sufficiently often" and "most" reducts should be chosen for a given data.
- Introduction of the notion of significance of attributes that allows by real values from the closed interval $[0, 1]$ to express the importance of this attribute in a decision table.

The Generalized Iterative algorithm based on the Rough Set approach (GIRS) has been developed by authors [17]. In the given algorithm we combined the discretization of quantitative attributes with the search of significant attributes. Therefore the process of searching a reduct is viewed as a search of attributes belonging to an approximated reduct. An approximated reduct is a generalization of a reduct where generalization is understood as the result of forming significant attributes on the basis of the rough set approach. The application of approximated reduct is very useful when processing incomplete and noisy data.

4.3 Noise Models

Assume that examples in a learning set contain noise, i.e., attribute values may be missing or be distorted. The noise arises due to following causes: incorrect measurement of the input parameter; wrong description of parameter values by an expert; the use of damaged measurement devices; and data lost in transmitting and storing the information [18]. We study two noise models:

1. The noise connected with the absence of attribute values (we could not receive an attribute value due to different causes). For each attribute A , a domain of admissible values includes the value "not known." Such a value corresponding to the situation when the true value of an attribute has been lost, is denoted by N (Not known). Thus, some examples of a learning sample U contain a certain quantity of attributes with the values "not known".

2. The noise connected with the distortion of certain attribute values in a learning sample. The true value in this case is replaced by one admissible, but wrong, value (the values are mixed) [18].

Further, we consider the work of the generalization algorithm in the presence of noise in input data. Our purpose is to assess a classification accuracy of examples in a control sample by increasing a noise level in this sample.

4.4 Algorithm GIRS for Noisy and Inconsistent Data

Let a sample with noise U' be given; moreover, let the attributes taking both discrete and continuous values be subject to distortions.

Building a system of production rules with examples having absent values leads to multivariant decisions. Therefore, we try to find the possibility for restoring these absent values. One of possible approach is the nearest neighbour method which was proposed for the classification of the unknown object o on the basis of consideration of several objects with the known classification nearest to it. The decision on the assignment of the object o to one or another class is made by information analysis on whether these nearest neighbours belong to one or another class. We can use even a simple count of votes to do this. The given method is implemented in the algorithm of restoring that was considered in detail in [19].

The above mentioned algorithm GIRS has been used to research the effect of a noise on forming generalized rules and on classification accuracy of test examples. Entering a noise, especially a distortion of attribute values in a learning sample, was performed on a decision attribute, what led to appearing inconsistent examples. To restore unknown values, the methods of nearest neighbours and choice of average are used. We have realized the algorithm GIRS including the RECOVERY algorithm. The RECOVERY algorithm is used at the presence of examples containing a noise of the type “*absent values*”. The RECOVERY algorithm is based on the nearest neighbour method which was proposed for the classification of the unknown object o on the basis of consideration of several objects with the known classification nearest to it. The detail overview of this algorithm is given in [19]. The decision on the assignment of the object o to one or another class is made by information analysis on whether these nearest neighbours belong to one or another class. We can use even a simple count of votes to do this. Since the solution in this method explicitly depends on the object quantity, we shall call this method as the search method of k nearest neighbours. The parameter defining a number of nearest objects, on basis of which a possible value of an unknown attribute is found, is called “the nearest neighbour number”.

Below, we present the pseudo code of the GIRS algorithm.

Algorithm GIRS

Given: table $S = (U, A \cup \{d\})$

Obtain: decision rules: R

Beginning

Obtaining S **Select** *noise model (absent values or distorted values), noise level*, one of two noise entering types

If there are “*absent values*” in S **then use** RECOVERY algorithm

Build A set of equivalence classes $\{X_1^A, X_2^A, \dots, X_m^A\}$

Build a lower approximation of set X and an upper approximation of set X

Find decision-relative reduct of table $S = (U, A \cup \{d\})$

Find the set of conditions:

$Pos(X)$ = lower approximation of set X ; $BND(X)$ = upper approximation of set $X \setminus$ lower approximation of set X ; $NEG(X) = \neg(\text{upper approximation of set } X)$

Build R – the set of classification rules in form:


```

IF  $Pos(X)$  THEN  $X$ ,
IF  $Neg(X)$  THEN  $\neg X$ ,
IF  $Bnd(X)$  THEN possible  $X$ 
Output decision rules  $R$ 
end
End of GIRS

```

4.5 Experimental Results and Analysis

Research of noise influence on the algorithm work was produced at the following conditions.

- There were used two noise models.
- Noise was entered in a learning set.
- A decision attribute was undergone by noise influence (by a decision attribute we understand such attribute that defines a class of each example).

A generalization model based on production rules was built on the noisy learning set. Built production rules were used for the classification of test sets that do not contain noise.

In table 1, there is given the results obtained for the "distortion" noise model. Noise of 5, 10, 15, 20 and 25 percent is introduced in a decision attribute.

We present experiment results carried out for data groups from the known collection of the test data sets of the California University of Informatics and Computer Engineering "UCI Machine Learning Repository" [20].

Data sets	No noise	Noise 5%	Noise 10%	Noise 15%	Noise 20%	Noise 25%
Monks-1	100	96,60	93,06	83,3	79,75	68,46
Monks-2	74,31	71,64	65,86	65,62	61,92	63,31
Monks-3	94,44	92,48	88,89	86,92	87,5	77,2

Table 1: Classification accuracy in the presence of "distortion of values" noise.

Data sets	No noise	Noise 5%	Noise 10%	Noise 15%	Noise 20%	Noise 25%
Monks-1	100	99,64	95,83	97,38	96,43	87,04
Monks-2	74,31	72,53	70,22	69,56	71,76	68,06
Monks-3	94,44	94,44	91,59	92,51	90,01	92,55

Table 2: Classification accuracy in the presence of "absence of values" noise.

In tables 1 and 2, the test results are presented, from which we can see that if we introduce the noise for up to 25% in the decision attribute of a learning sample, the decrease of the quality level of the formed rules was observed, but it is worth noting that a noise level was high enough (more than 10%). We should take into consideration that the accuracy of classification under the noise level more than 20% is greatly decreased. It is related with the appearance of the great number of inconsistent samples in learning sets.

Obviously, with the growth of a noise level, the classification accuracy of the test examples is decreased, but it is difficult to judge about the monotonicity or non-monotonicity of noise influence on the classification quality level.

The given experiments show that the developed algorithm allows to reduce time for searching the significant attributes particularly due to combination with a discretization process. For all data sets taken into the comparison, this algorithm has shown the classification accuracy that is not worse than the accuracy of generalization algorithms, such as C 4.5, CN2, J48, SMO, but in some cases

surpasses it [21], [22], [13]. The average accuracy of classification is approximately 88.9%. Comparison of different algorithms of classification accuracy of test samples with noise could be found in [13].

5 Conclusion

We've considered the problem of inconsistent information and given some approaches of dealing with such data in IDSSs. It was proposed to use argumentation in the decision unit and methods of rough sets for the generalization problem.

Argumentation systems based on defeasible reasoning could be used in many intelligent systems that use the first order logic as means of knowledge representation. It was shown that it is a powerful tool for dealing with inconsistent data. Nevertheless, the problem of justification degrees remains open. Implementation of different mechanisms of calculating justification degrees, including methods of fuzzy logic, is a topic of a future work.

The noise models in DB including absent or distorted attribute values in learning samples were considered. The algorithm GIRS allowing to process learning sets that contain examples with unknown or distorted values has been suggested. The system of building generalized notions using the obtained theoretical results has been implemented. The obtained results have shown that the algorithm GIRS essentially enhances the classification accuracy of examples in the presence of noise in data.

References

- [1] V. Vagin and A. Eremeev, "Methods and Tools for Modelling Reasoning in Diagnostic Systems," in *Proceedings of the 11th International Conference on Enterprise Information Systems*, Milan, 2009.
- [2] P. Besnard and A. Hunter, *Elements of Argumentation*, Cambridge: MIT Press, 2008, p. 298.
- [3] A. Bondarenko, P. Dung, R. Kowalski, F. Toni and A. Bondarenko, "An Abstract Argumentation-Theoretic Framework for Defeasible Reasoning," *Artificial Intelligence*, vol. 93, no. 1-2, pp. 63-101, 1997.
- [4] F. Lin and Y. Shoham, "Argument Systems. A Uniform Basis for Nonmonotonic Reasoning," San Mateo, 1989.
- [5] G. Vreeswijk, "Abstract Argumentation Systems," *Artificial Intelligence*, vol. 90, pp. 225-279, 1997.
- [6] J. Pollock, "Oscar – a General Purpose Defeasible Reasoner," *Journal of Nonclassical Logics*, vol. 6, pp. 89-113, 1996.
- [7] N. Gorogiannis and A. Hunter, "Instantiating abstract argumentation with classical logic arguments: Postulates and properties," *Artificial Intelligence*, vol. 175, no. 9–10, pp. 1479-1497, 2011.
- [8] D. Walton, *Methods of Argumentation*, Cambridge MA: Cambridge University Press, 2013, p. 317.

- [9] V. N. Vagin, E. Golovina, A. Zagoryanskaya and M. Fomina, Exact and Plausible Inference in Intelligent Systems (in Russian), V. Vagin and D. Pospelov, Eds., Moscow: FizMatLit, 2008, p. 714.
- [10] V. Vagin and O. Morosin, "Modeling defeasible reasoning for argumentation," in *Proceedings of BRICS-CCI'2013 – 1st BRICS Countries Congress on Computational Intelligence and CBIC'2013*, Recife, 2013.
- [11] G. Betz, "On Degrees of Justification," *Erkenntnis*, vol. Volume, no. 2, pp. 237-272, 2012.
- [12] R. Haenni, J. Kohlas and N. Lehmann, "Probabilistic Argumentation Systems," in *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 5: Algorithms for Uncertainty and Defeasible Reasoning, Dordrecht, Kluwer, 1999, p. 221–287.
- [13] M. Mannino, Y. Yang and Y. Ryu, "Classification Algorithm Sensitivity to Training Data with Non Representative Attribute Noise," *Decision Support Systems*, vol. 46, no. 3, pp. 743-751, 2009.
- [14] J. Komorowski, Z. Pawlak, L. Polkowski and A. Skowron, *Rough Sets: A Tutorial*, Springer-Verlag, 1999.
- [15] Z. Pawlak, "Rough sets and intelligent data analysis," *Information Sciences*, vol. 147, no. 1, pp. 1-12, 2002.
- [16] S. Nguyen and H. Nguyen, "Some efficient algorithms for rough set methods," in *Proc. of the Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Spain, 1996.
- [17] V. Vagin, M. Fomina and A. Kulikov, "The Problem of Object Recognition in the Presence of Noise in Original Data," in *10th Scandinavian Conference on Artificial Intelligence SCAI*, 2008.
- [18] V. Mookerjee, M. Mannino and R. Gilson, "Improving the Performance Stability of Inductive Expert Systems under Input Noise," *Information Systems Research*, vol. 6, no. 4, pp. 328-356, 1995.
- [19] V. Vagin and M. Fomina, "Problem of Knowledge Discovery in Noisy Databases," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 3, p. 135 – 145, 2011.
- [20] C. Merz and M. P., "UCI Repository of Machine Learning Datasets. Information and Computer Science University of California," 1998. [Online]. Available: <http://archive.ics.uci.edu/ml/>. [Accessed 10 03 2014].
- [21] R. Quinlan, *C4.5: Programs for Machine Learning*, San Francisco: Morgan Kaufmann Publishers Inc., 1993.
- [22] P. Clark and R. Boswell, *Rule Induction with CN2: Some Recent Improvements*, Springer-Verlag, 1991, pp. 151-163.